

# Eigenes Histogramm

## Eigenes Histogramm

Es soll ein eigenes Histogramm erzeugt werden. Der Dateiname für das Skript ist `myhistogram.R`.

### Funktion `myhistogram`

Programmieren Sie in R die Funktion `myhistogram`, die als Parameter `x` einen Vektor aus Zahlen erhält. Die Zahlen werden in `n` Intervalle einsortiert, und es wird gezählt, wie oft eine Zahl in einem Intervall vorkommt. Der Rückgabewert ist eine Liste mit den Einträgen `borders`, die die  $n + 1$  Intervallgrenzen enthalten und `counts`, der die Anzahlen enthält.

- Die  $n$  Intervalle sollen gleich groß sein ( $\Delta b$ ), d.h. für die Intervallgrenzen  $b_1, b_2, \dots, b_{n+1}$  gilt  $\frac{b_{i+1} - b_i}{n} = \Delta b$  für  $i = 1, 2, \dots, n$ .
- Die äußeren Grenzen  $b_1$  und  $b_n$  sollen als optionale Parameter `min` und `max` an die Funktion übergeben werden. Werte aus `x`, die zu keinem Intervall gehören, sollen ignoriert werden. Es wird aber eine Warnung ausgegeben, die sagt, welche Zahlen außerhalb des Bereichs liegen.
- Eine Zahl  $z$  gehört zum  $i$ -ten Intervall, falls  $b_i \leq z < b_{i+1}$  gilt.

Bis auf `x` sollen alle Parameter optional sein. Überlegen Sie sinnvolle Default-Werte.

Es ist natürlich **nicht** erlaubt, in der eigenen Funktion andere Funktionen zu nutzen, die ein Histogramm erzeugen.

Hier ein Beispiel:

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)

l = myhistogram(x, 5, -3)

print(l)
print(l$borders)
print(l$counts)
```

## Beispieldaten

Hier zunächst zwei Beispiele.

### Beispiel 1

Es wird eine Warnung ausgegeben:

```
## Warning in myhistogram(x, n = 10, min = -5, max = 6): Zahl(en) außerhalb
## Intervallgrenzen: 6
```

```
# Anmerkung: Bei VSCode auf macOS war es für 'Knit' nötig,
# in jedem Code-Chunk die source erneut anzugeben...
```

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)
myhistogram(x, n = 10, min = -5, max = 6)
```

```
## Warning in myhistogram(x, n = 10, min = -5, max = 6): Zahl(en) außerhalb
## Intervallgrenzen: 6

## $borders
## [1] -5.0 -3.9 -2.8 -1.7 -0.6  0.5  1.6  2.7  3.8  4.9  6.0
##
## $counts
## [1] 4 3 3 4 3 3 4 3 3 3

solution1 = list(
  borders = c(-5.0, -3.9, -2.8, -1.7, -0.6, 0.5, 1.6, 2.7, 3.8, 4.9, 6.0),
  counts = c(4, 3, 3, 4, 3, 3, 4, 3, 3, 3)
)
expected_warning = 'Zahl(en) außerhalb Intervallgrenzen: 6'
```

## Beispiel 2

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)
myhistogram(x, n = 5, min = -10, max = 10)

## $borders
## [1] -10 -6 -2  2  6 10
##
## $counts
## [1]  0  9 12 12  1

solution2 = list(borders = c(-10, -6, -2, 2, 6, 10),
  counts = c(0, 9, 12, 12, 1))
```

## Beispiel 3

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x3 = c(10, 12, 15, 20, 25, 30, 35, 40, 45)
myhistogram(x3, n = 5, min = 5, max = 30)

## Warning in myhistogram(x3, n = 5, min = 5, max = 30): Zahl(en) außerhalb
## Intervallgrenzen: 30, 35, 40, 45

## $borders
## [1]  5 10 15 20 25 30
##
## $counts
## [1] 0 2 1 1 1

solution3 = list(borders = c(5, 14, 23, 32, 41, 50),
  counts = c(2, 2, 2, 2, 1))
expected_warning = 'Zahl(en) außerhalb Intervallgrenzen: 30, 35, 40, 45'
```

## Beispiel 4

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x2 = c(-2, -1, 0, 1, 2, 3, 4, 5)
myhistogram(x2, n = 4, min = -3, max = 6)
```

```
## $borders
## [1] -3.00 -0.75  1.50  3.75  6.00
##
## $counts
## [1] 2 2 2 2

solution4 = list(borders = c(-3, -0.75, 1.50, 3.75, 6.00),
                 counts = c(2, 2, 2, 2))
```

## Beispiel 5

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x5 = rep(0, 10)
myhistogram(x5, n = 4, min = -3, max = 6)

## $borders
## [1] -3.00 -0.75  1.50  3.75  6.00
##
## $counts
## [1]  0 10  0  0

solution5 = list(borders = c(-3, -1.5, 0, 1.5, 3, 4.5, 6),
                 counts = c(0, 10, 0, 0, 0, 0))
```

## Barplot

Nutzen Sie Ihre Funktion `myhistogram` und erzeugen Sie einen Barplot mit *ggplot*. Die *x*-Achse zeigt dabei die Mitte des Intervalls und die *y*-Achse die Anzahl der Elemente in dieser Klasse.

Tipp: Der Parameter `stat` von `geom_bar` ist wichtig.

Vervollständigen Sie den Chunk. Die Kommentare sollen zu Anweisungen umgewandelt werden:

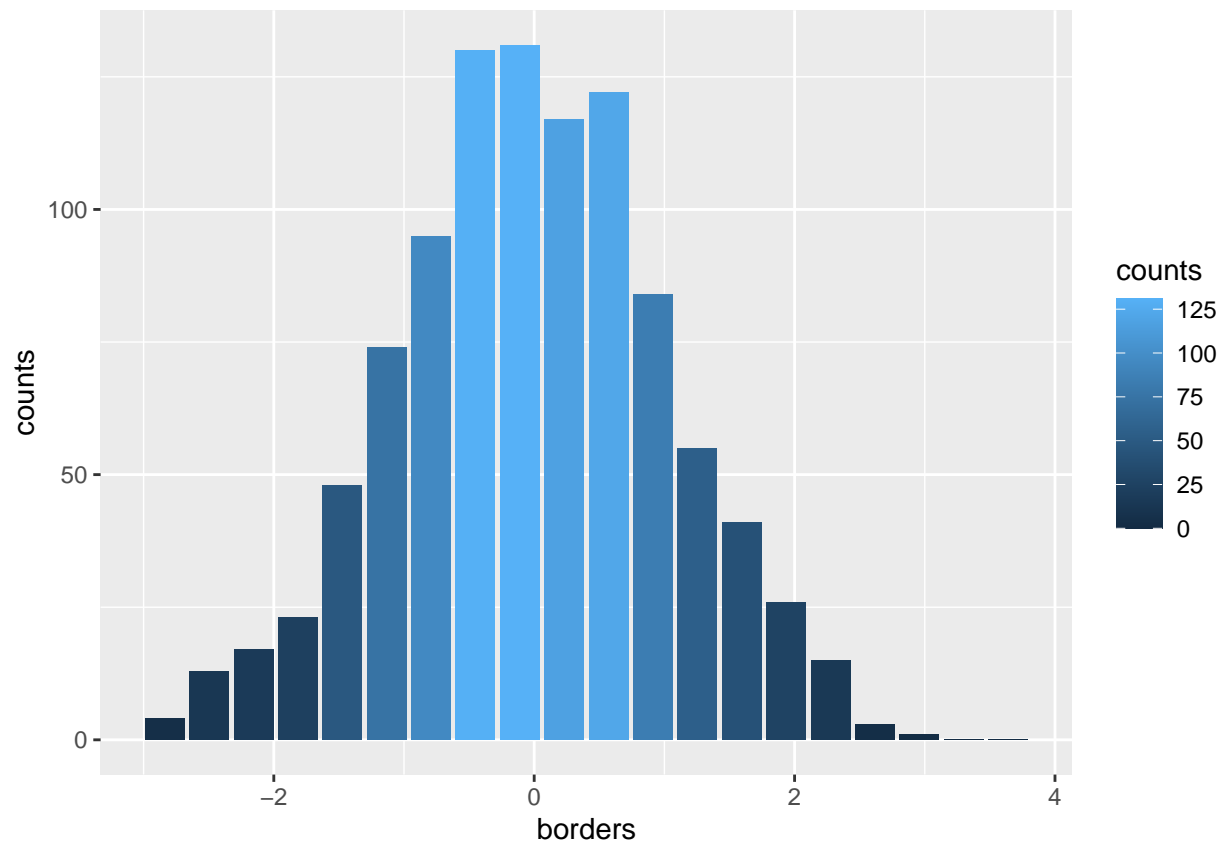
```
source("myhistogram.R")
library("ggplot2")
library("knitr")
set.seed(1)
x = rnorm(0, 1, n = 1000)
h = myhistogram(x, n = 20)

## Warning in myhistogram(x, n = 20): Zahl(en) außerhalb Intervallgrenzen:
## 3.81027668071067

mittelpunkte = (h$borders[-1] + h$borders[-length(h$borders)]) / 2

df = data.frame(borders = mittelpunkte, counts = h$counts[])

ggplot(df) +
  geom_bar(aes(x = borders, y = counts, fill = counts), stat = "identity")
```



```
ggsave(filename = 'myhistogram.jpg')
```

```
## Saving 6.5 x 4.5 in image
```

# Plot

## Visualisierung von Datensätzen

In diesem Abschnitt sollen alle Graphiken mit *ggplot* und alle Tabellen mit **kable** erstellt werden.

### Körpergewicht und Gehirngewicht bei Säugetieren

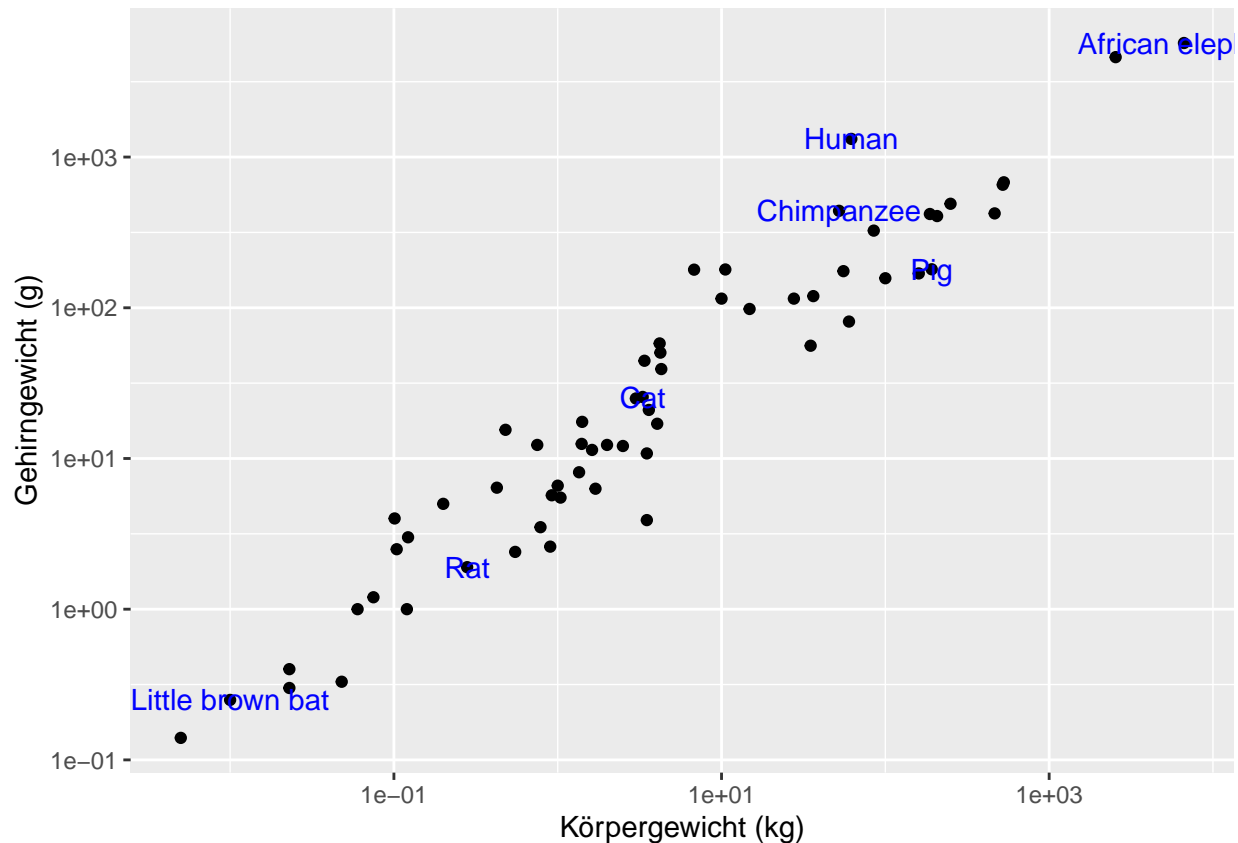
Nutzen Sie den Datensatz `MASS::mammals`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist.

#### Körpergewicht vs. Gehirngewicht

Erzeugen Sie diese Graphik, indem Sie den nachfolgenden Chunk vervollständigen. Die gezeigten Tiernamen sind Pig, Rat, African elephant, Chimpanzee, Cat, Human, Little brown bat.

Tipp: Sie dürfen (und sollen) weitere Libraries nutzen, wenn diese hilfreich sind.

```
selected_mammals = mammals[c("Pig", "Rat", "African elephant", "Chimpanzee", "Cat",  
"Human", "Little brown bat"), ] # rausgefilterte Tiere für Label  
  
ggplot(mammals) +  
  geom_point(aes(x = body, y = brain)) + # Erstellt Scatterplot  
  scale_x_continuous("Körpergewicht (kg)", labels = scientific, trans = "log10") +  
  scale_y_continuous("Gehirngewicht (g)", labels = scientific, trans = "log10") +  
  geom_text(data = selected_mammals, aes(x = body, y = brain, label =  
    rownames(selected_mammals)), color = "blue") # erstellt Label
```



### Gehirn- zu Körpergewicht-Verhältnis

Geben Sie diejenigen 10 Tiere als Tabelle im Notebook aus, die das größte Gehirn- zu Körpergewicht-Verhältnis  $r$  haben.

Die Liste soll nach  $r$  absteigend sortiert sein und den Tiernamen und  $r$  enthalten.

Vervollständigen Sie diesen Chunk:

```
```r
df_ratio_desc = mammals |> mutate(r = brain*0.001/body) |> arrange(desc(r)) |> dplyr::select(r)
kable(head(df_ratio_desc, 10))
```

	r
Ground squirrel	0.0396040
Owl monkey	0.0322917
Lesser short-tailed shrew	0.0280000
Rhesus monkey	0.0263235
Little brown bat	0.0250000
Galago	0.0250000
Mole rat	0.0245902
Tree shrew	0.0240385
Human	0.0212903
Mouse	0.0173913

Geben Sie nun – wie eben – diejenigen 10 Tiere als Tabelle aus, die das **kleinste** Gehirn- zu Körpergewicht-Verhältnis  $r$  haben. Die Liste soll nach  $r$  absteigend sortiert sein.

Vervollständigen Sie diesen Chunk:

```
df_ratio_asc = mammals |> mutate(r = brain*0.001/body) |> arrange(r) |> dplyr::select(r)
knitr::kable(head(df_ratio_asc, 10))
```

	r
African elephant	0.0008584
Cow	0.0009097
Pig	0.0009375
Brazilian tapir	0.0010563
Water opossum	0.0011143
Horse	0.0012572
Giraffe	0.0012854
Giant armadillo	0.0013500
Jaguar	0.0015700
Kangaroo	0.0016000

## Blutdruckveränderung bei Medikamentengabe im Tierversuch

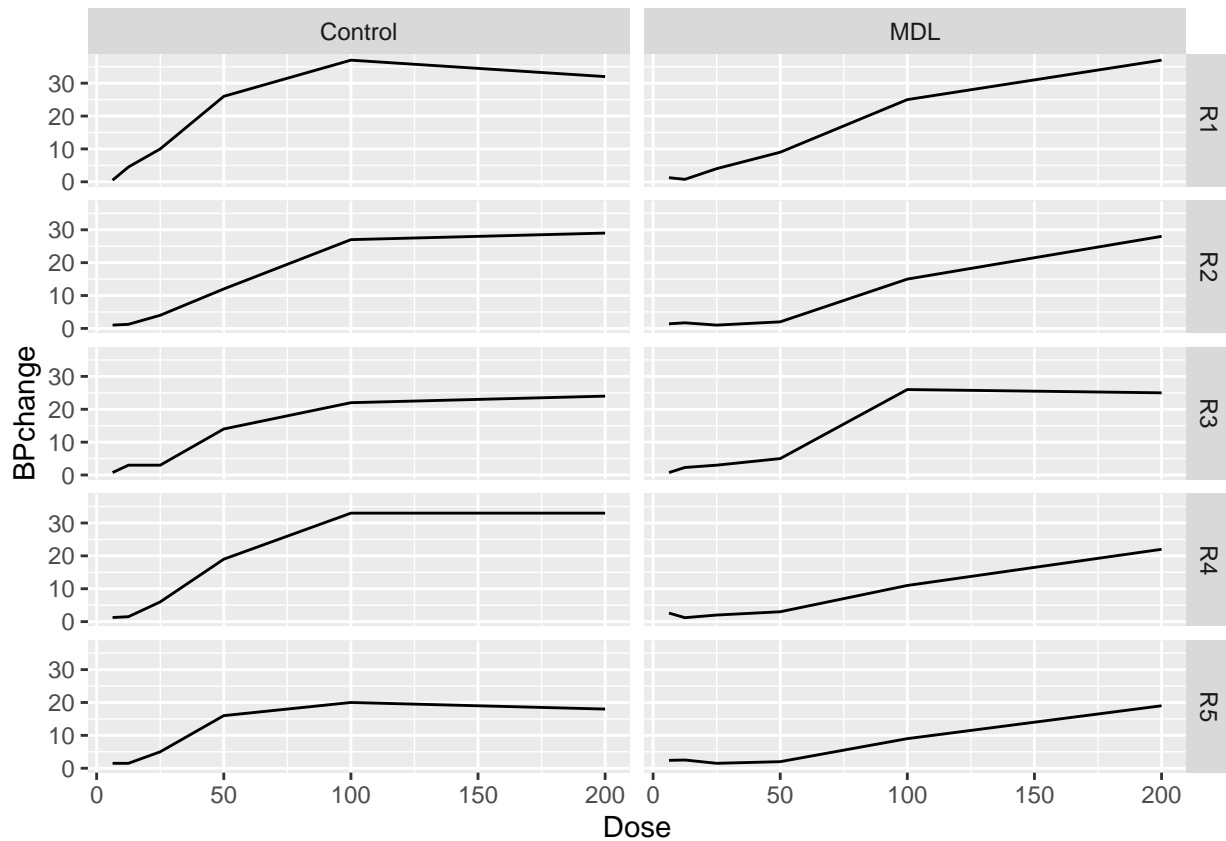
Nutzen Sie den Datensatz `MASS::Rabbit`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist.

### Überblick über Verlauf bei allen Kaninchen

Plotten Sie im folgenden Chunk den Verlauf der Blutdruckveränderung ( $y$ -Achse) bei gegebener Dosis Phenylbiguanide ( $x$ -Achse). Dies soll in einem Diagramm mit Unterdiagrammen erfolgen: ein Unterdiagramm zeigt den Verlauf für je ein Kaninchen und der Behandlung (Placebo oder MDL 72222).

```
data(Rabbit)

ggplot(Rabbit, aes(x = Dose, y= BPchange)) +
  geom_line() +
  facet_grid( Rabbit$Animal ~ Rabbit$Treatment) # erstelle unterdiagramm
```



### Boxplots der Blutdruckänderung je Dosis

Erzeugen Sie ein Diagramm, das in zwei Unterdiagrammen für die Placebo- und die MLD-Gruppe Boxplots erstellt. Die Boxplots geben die Verteilung der Blutdruckänderung je Dosis an. In Anlehnung an das obige Diagramm sollen die Boxplots vertikal ausgerichtet sein.

```
control = Rabbit[Rabbit$Treatment == 'Control',]
mdl = Rabbit[Rabbit$Treatment == 'MDL',]

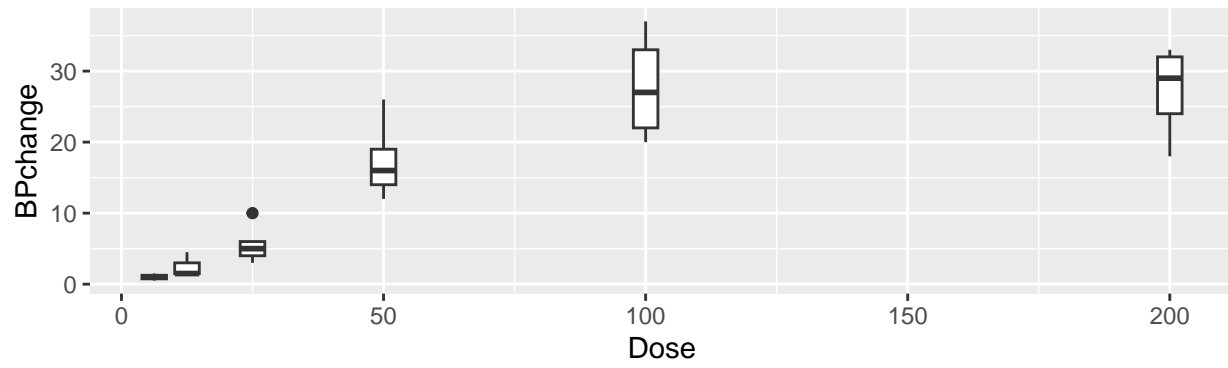
p1 = ggplot(data = control) +
  geom_boxplot(aes(group=Dose, y=BPchange, x=Dose)) +
  ggtitle('Verteilung der Blutdruckänderung pro Placebo')

p2 = ggplot(data = mdl) +
  geom_boxplot(aes(group=Dose, y=BPchange, x=Dose)) +
  ggtitle('Verteilung der Blutdruckänderung pro Dosis MDL')

grid.arrange(p1,p2, nrow=2)
```



Verteilung der Blutdruckänderung pro Placebo



Verteilung der Blutdruckänderung pro Dosis MDL



# Impfungen

## Covid19: Impffortschritt in Deutschland

In dieser Aufgabe geht es um den Verlauf der Corona-Impfungen in Deutschland. Die folgenden URLs enthalten Daten ab 2020:

- [https://impfdashboard.de/static/data/germany\\_vaccinations\\_timeseries\\_v2.tsv](https://impfdashboard.de/static/data/germany_vaccinations_timeseries_v2.tsv)
- [https://impfdashboard.de/static/data/germany\\_deliveries\\_timeseries\\_v2.tsv](https://impfdashboard.de/static/data/germany_deliveries_timeseries_v2.tsv)
- [https://impfdashboard.de/static/data/germany\\_vaccinations\\_by\\_state.tsv](https://impfdashboard.de/static/data/germany_vaccinations_by_state.tsv)

Sie sind der Webseite <https://impfdashboard.de> entnommen.

### Einlesen der Daten

Lesen Sie die drei Dateien je in einen Data Frame ein mit den Variablennamen:

- `vacc`
- `deliv`
- `vaccState`

Wandeln Sie die Datums- und Zeitangaben von einem String in ein R-Datumsobjekt um. Geben Sie die ersten drei Zeilen und Spalten dieser Data Frames aus.

```
vacc = read.csv(url('https://impfdashboard.de/static/data/germany_vaccinations_timeseries_v2.tsv'),
sep='\t')
vacc$date = as.Date(vacc$date, "%Y-%m-%d")
print(head(vacc, 3))
```

```
##           date dosen_kumulativ dosen_biontech_kumulativ
## 1 2020-12-27           24427           24418
## 2 2020-12-28           42500           42489
## 3 2020-12-29           92729           92717
##   dosen_biontech_erst_kumulativ dosen_biontech_zweit_kumulativ
## 1                        24418                        0
## 2                        42489                        0
## 3                        92717                        0
##   dosen_biontech_dritt_kumulativ dosen_biontech_viert_kumulativ
## 1                        0                        0
## 2                        0                        0
## 3                        0                        0
##   dosen_biontech_fuenft_kumulativ dosen_biontech_sechst_kumulativ
## 1                        0                        0
## 2                        0                        0
## 3                        0                        0
##   dosen_moderna_kumulativ dosen_moderna_erst_kumulativ
## 1                        8                        8
## 2                        9                        9
## 3                        9                        9
##   dosen_moderna_zweit_kumulativ dosen_moderna_dritt_kumulativ
## 1                        0                        0
```

## 2	0	0
## 3	0	0
##	dosen_moderna_viert_kumulativ	dosen_moderna_fuenft_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_moderna_sechst_kumulativ	dosen_astra_kumulativ
## 1	0	1
## 2	0	2
## 3	0	3
##	dosen_astra_erst_kumulativ	dosen_astra_zweit_kumulativ
## 1	1	0
## 2	2	0
## 3	3	0
##	dosen_astra_dritt_kumulativ	dosen_astra_viert_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_astra_fuenft_kumulativ	dosen_astra_sechst_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_johnson_kumulativ	dosen_johnson_erst_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_johnson_zweit_kumulativ	dosen_johnson_dritt_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_johnson_viert_kumulativ	dosen_johnson_fuenft_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_johnson_sechst_kumulativ	dosen_novavax_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_novavax_erst_kumulativ	dosen_novavax_zweit_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_novavax_dritt_kumulativ	dosen_novavax_viert_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_novavax_fuenft_kumulativ	dosen_novavax_sechst_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0
##	dosen_valneva_kumulativ	dosen_valneva_erst_kumulativ
## 1	0	0
## 2	0	0
## 3	0	0

```

##   dosen_valneva_zweit_kumulativ  dosen_valneva_dritt_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_valneva_viert_kumulativ  dosen_valneva_fuenft_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_valneva_sechst_kumulativ  dosen_biontech_bivalent_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_biontech_bivalent_erst_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_biontech_bivalent_zweit_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_biontech_bivalent_dritt_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_biontech_bivalent_viert_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_biontech_bivalent_fuenft_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_biontech_bivalent_sechst_kumulativ  dosen_moderna_bivalent_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_moderna_bivalent_erst_kumulativ  dosen_moderna_bivalent_zweit_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_moderna_bivalent_dritt_kumulativ  dosen_moderna_bivalent_viert_kumulativ
## 1                               0                               0
## 2                               0                               0
## 3                               0                               0
##   dosen_moderna_bivalent_fuenft_kumulativ
## 1                               0
## 2                               0
## 3                               0
##   dosen_moderna_bivalent_sechst_kumulativ  dosen_erst_kumulativ
## 1                               0                24427
## 2                               0                42500
## 3                               0                92729
##   dosen_zweit_kumulativ  dosen_dritt_kumulativ  dosen_viert_kumulativ
## 1                      0                      0                      0

```

## 2	0	0	0
## 3	0	0	0
##	dosen_fuenft_kumulativ	dosen_sechst_kumulativ	dosen_differenz_zum_vortag
## 1	0	0	24427
## 2	0	0	18073
## 3	0	0	50229
##	dosen_erst_differenz_zum_vortag	dosen_zweit_differenz_zum_vortag	
## 1	24427		0
## 2	18073		0
## 3	50229		0
##	dosen_dritt_differenz_zum_vortag	dosen_viert_differenz_zum_vortag	
## 1	0		0
## 2	0		0
## 3	0		0
##	dosen_fuenft_differenz_zum_vortag	dosen_sechst_differenz_zum_vortag	
## 1	0		0
## 2	0		0
## 3	0		0
##	dosen_vollstaendig_differenz_zum_vortag		
## 1	0		
## 2	0		
## 3	0		
##	dosen_erst_unvollstaendig_differenz_zum_vortag	personen_erst_kumulativ	
## 1		24427	24427
## 2		18073	42500
## 3		50229	92729
##	personen_voll_kumulativ	personen_auffrisch_kumulativ	
## 1	0	0	
## 2	0	0	
## 3	0	0	
##	personen_auffrisch2_kumulativ	personen_auffrisch3_kumulativ	
## 1	0	0	
## 2	0	0	
## 3	0	0	
##	personen_auffrisch4_kumulativ	impf_quote_erst	impf_quote_voll
## 1	0	0.00029	0
## 2	0	0.00051	0
## 3	0	0.00111	0
##	dosen_dim_kumulativ	dosen_kbv_kumulativ	indikation_alter_dosen
## 1	0	0	0
## 2	0	0	0
## 3	0	0	0
##	indikation_beruf_dosen	indikation_medizinisch_dosen	
## 1	0	0	
## 2	0	0	
## 3	0	0	
##	indikation_pflegeheim_dosen	indikation_alter_erst	indikation_beruf_erst
## 1	0	0	0
## 2	0	0	0
## 3	0	0	0
##	indikation_medizinisch_erst	indikation_pflegeheim_erst	indikation_alter_voll
## 1	0	0	0
## 2	0	0	0
## 3	0	0	0

```
##   indikation_beruf_voll indikation_medizinisch_voll indikation_pflegeheim_voll
## 1                0                0                0
## 2                0                0                0
## 3                0                0                0

deliv = read.csv(url('https://impfdashboard.de/static/data/germany_deliveries_timeseries_v2.tsv'),
  sep='\t')
deliv$date = as.Date(deliv$date, "%Y-%m-%d")
print(head(deliv, 3))

##           date impfstoff impfstofftyp region dosen einrichtung
## 1 2020-12-26 comirnaty      wildtyp  DE-BW  9750 impfzentren
## 2 2020-12-26 comirnaty      wildtyp  DE-BY  9750 impfzentren
## 3 2020-12-26 comirnaty      wildtyp  DE-BE  9750 impfzentren

vaccState = read.csv(url('https://impfdashboard.de/static/data/germany_vaccinations_by_state.tsv'),
  sep='\t')
print(head(vaccState, 3))

##      code vaccinationsTotal peopleFirstTotal peopleFullTotal peopleBoosterTotal
## 1  DE-BB           5080605           1723412           1633105           1417320
## 2  DE-BE           8691294           2902129           2766462           2358876
## 3 DE-BUND           549247            202150            190143            146098
```

## Verimpfte Impfdosen pro Tag

Es soll untersucht werden, wie oft welcher Impfstoff an welchem Tag verimpft wurde.

### Transformation

Der Data Frame `vacc` enthält leider keine Angaben, wie oft ein Impfstoff eines Herstellers täglich verabreicht wurde. Erzeugen Sie aus `vacc` einen neuen Data Frame `vacc2`, der die folgende Struktur hat:

Table 1: Neue Struktur: Data Frame `vacc2`.

Datum	Hersteller	Impfdosen pro Tag
09.04.21	biontech	123456
09.04.21	moderna	12345
...	...	...

Wie Sie die Impfstoffe (biontech, moderna, astra) nennen, bleibt Ihnen überlassen – solange die Bezeichnungen konsistent und schlüssig sind.

Geben Sie die letzten Zeilen von `vacc2` als `kable` aus. Tipp: `tail` gibt die letzten Zeilen eines Data Frames an (analog zu `head`).

```
vacc2 = vacc |> mutate(
  biontech = dosen_biontech_kumulativ - lag(dosen_biontech_kumulativ, 1, 0),
  moderna = dosen_moderna_kumulativ - lag(dosen_moderna_kumulativ, 1, 0),
  astra = dosen_astra_kumulativ - lag(dosen_astra_kumulativ, 1, 0)) |>
  select(date, biontech, moderna, astra) |> pivot_longer(cols=c(2,3,4))

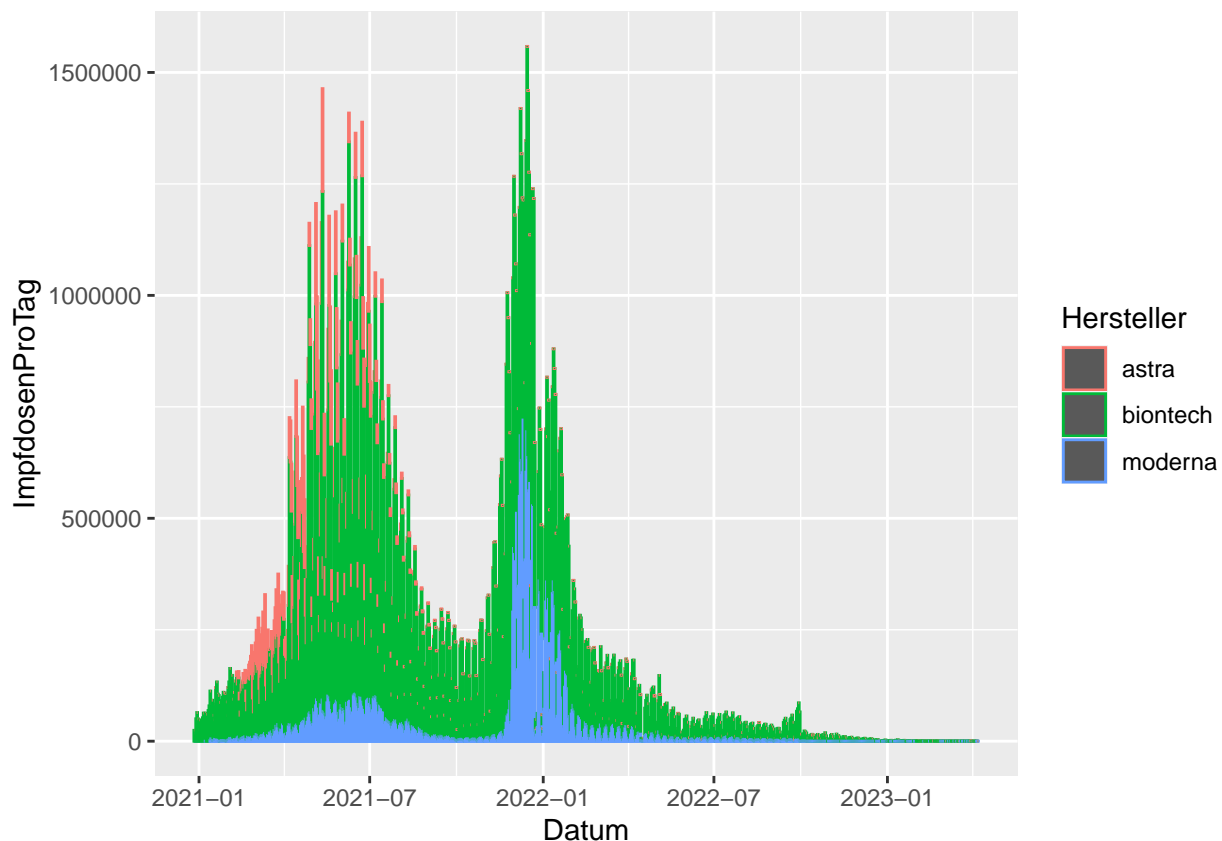
names(vacc2)[names(vacc2) == "date"] <- "Datum"
names(vacc2)[names(vacc2) == "name"] <- "Hersteller"
names(vacc2)[names(vacc2) == "value"] <- "ImpfdosenProTag"
kable(tail(vacc2,10))
```

Datum	Hersteller	ImpfdosenProTag
2023-04-04	astra	0
2023-04-05	biontech	277
2023-04-05	moderna	2
2023-04-05	astra	0
2023-04-06	biontech	281
2023-04-06	moderna	37
2023-04-06	astra	0
2023-04-07	biontech	15
2023-04-07	moderna	39
2023-04-07	astra	0

### Plot der täglichen Impfdosen nach Hersteller

Plotten Sie mit *ggplot* den Verlauf der täglichen Impfdosen für jeden Hersteller. Die *x*-Achse zeigt das Datum und die *y*-Achse die Anzahl der Impfdosen pro Tag. Überlegen Sie, welcher Diagrammtyp dafür am besten geeignet ist.

```
ggplot(vacc2) +
  geom_bar(aes(x=Datum, y=ImpfdosenProTag, group=Hersteller, color=Hersteller),
    stat='identity')
```



### Zeitverzug Auslieferung bis Verimpfung

Es soll untersucht werden, wie schnell gelieferte Impfmengen der einzelnen Impfstoffe auch verimpft wurden. Es bietet sich dafür an, die akkumulierten Impfdosen mit den akkumulierten Impflieferungen zeitlich plotten.

Je größer die Lücke zwischen der Liefermenge und der Impfungen ist, desto mehr Impfstoff blieb liegen. Die Graphik soll Angaben für ganz Deutschland und nicht für die einzelnen Bundesländer zeigen.

Hinweis: Auch hier ist eine Vorverarbeitung der Daten nötig.

Plotten Sie dies mit *ggplot*:

```
used = vacc |> select(date, dosen_kumulativ)
delivered = aggregate(deliv$dosen, by=list(date=deliv$date), FUN=sum) |> rename(dosen=x)
delivered$dosen = delivered$dosen |> cumsum()

ggplot() +
  geom_bar(data=delivered, aes(x=date, y=dosen), stat='identity', color='red', ) +
  geom_bar(data=used, aes(x=date, y=dosen_kumulativ), stat='identity')
```

